Motivation

- **Rowhammer**
  - Default refresh window of 64 ms
Motivation

- **Rowhammer**
  - Default refresh window of 64 ms
- **Error Corretting Code (ECC)**
  - Correct only one flip
Motivation

- **Rowhammer**
  - Default refresh window of 64 ms

- **Error Correting Code (ECC)**
  - Correct only one flip

- **Targeted Row Refresh (TRR)**
  - Refresh direct neighbours hammering rows
  - Exhaustion with multi-sided patterns [2, 1]
Motivation

- **Rowhammer**
  - Default refresh window of 64 ms
- **Error Correting Code (ECC)**
  - Correct only one flip
- **Targeted Row Refresh (TRR)**
  - Refresh direct neighbours hammering rows
  - Exhaustion with multi-sided patterns [2, 1]
- Would perfect TRR fix Rowhammer attacks?

Andreas Kogler (@0xhilbert)  Jonas Juffinger (@notimaginary...)
Observed Flips

- Short answer: No

Hammering with three rows between the aggressors causes flips on LPDDR4x commodity devices. 5 out of 7 mobile devices affected. With active TRR and on-chip ECC.

Is this Distance-2 Rowhammer?

Andreas Kogler (@0xhilbert) Jonas Juffinger (@notimaginary_)
Observed Flips

- Short answer: No
- Hammering with three rows between the aggressors
  - Causes flips on LPDDR4x commodity devices
  - 5 out of 7 mobile devices affected
  - With active TRR and on-chip ECC
Observed Flips

• Short answer: No
• Hammering with three rows between the aggressors
  • Causes flips on LPDDR4x commodity devices
  • 5 out of 7 mobile devices affected
  • With active TRR and on-chip ECC
• Is this Distance-2 Rowhammer?
• What is the root cause?
## FPGA Experiments

<table>
<thead>
<tr>
<th></th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>Far Aggressor</td>
<td>$\mathcal{F}_+$</td>
</tr>
<tr>
<td>Near Aggressor</td>
<td>$\mathcal{N}_+$</td>
</tr>
<tr>
<td>Victim</td>
<td>$\mathcal{V}$</td>
</tr>
<tr>
<td>Near Aggressor</td>
<td>$\mathcal{N}_-$</td>
</tr>
<tr>
<td>Far Aggressor</td>
<td>$\mathcal{F}_-$</td>
</tr>
</tbody>
</table>

- FPGA setup
  - Control DIMM via FPGA
  - Full control over the refreshes
  - Deactivated TRR
  - No need for data retention
### FPGA Experiments - Distance 1

<table>
<thead>
<tr>
<th>Role</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>Far Aggressor</td>
<td>$F_+$</td>
</tr>
<tr>
<td>Near Aggressor</td>
<td>$N_+$</td>
</tr>
<tr>
<td>Victim</td>
<td>$V$</td>
</tr>
<tr>
<td>Near Aggressor</td>
<td>$N_-$</td>
</tr>
<tr>
<td>Far Aggressor</td>
<td>$F_-$</td>
</tr>
</tbody>
</table>

- **Distance-1**
  - $(N_+ \rightarrow N_-)^\infty$
  - *Classic* double-sided Rowhammer

Andreas Kogler (@0xhilbert)  Jonas Juffinger (@notimaginary_)
FPGA Experiments - Distance 1

- **Distance-1**
  - \((\mathcal{N}_+ \rightarrow \mathcal{N}_-)\)\(^\infty\)
  - *Classic* double-sided Rowhammer
- **First** flip after:
  - 18 000 hammers in 1.2 ms
FPGA Experiments - Distance 1

- **Distance-1**
  - \((N_+ \rightarrow N_-)^\infty\)
  - *Classic* double-sided Rowhammer

- **First** flip after:
  - 18,000 hammers in 1.2 ms
  - ✓ *Within* the refresh window
  - ✗ *Mitigated* by TRR
• **Distance-2**
  - \((\mathcal{F}_+ \rightarrow \mathcal{F}_-)^\infty\)
  - Distance two double-sided Rowhammer

<table>
<thead>
<tr>
<th>Far Aggressor</th>
<th>((\mathcal{F}_+))</th>
</tr>
</thead>
<tbody>
<tr>
<td>Near Aggressor</td>
<td>( (\mathcal{N}_+))</td>
</tr>
<tr>
<td>Victim</td>
<td>((\mathcal{V}))</td>
</tr>
<tr>
<td>Near Aggressor</td>
<td>( (\mathcal{N}_-))</td>
</tr>
<tr>
<td>Far Aggressor</td>
<td>((\mathcal{F}_-))</td>
</tr>
</tbody>
</table>
**FPGA Experiments - Distance 2**

<table>
<thead>
<tr>
<th>Far Aggressor</th>
<th>( \mathcal{F}_+ )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Near Aggressor</td>
<td>( \mathcal{N}_+ )</td>
</tr>
<tr>
<td>Victim</td>
<td>( \mathcal{V} )</td>
</tr>
<tr>
<td>Near Aggressor</td>
<td>( \mathcal{N}_- )</td>
</tr>
<tr>
<td>Far Aggressor</td>
<td>( \mathcal{F}_- )</td>
</tr>
</tbody>
</table>

- **Distance-2**
  - \( (\mathcal{F}_+ \rightarrow \mathcal{F}_-)^\infty \)
  - Distance two double-sided Rowhammer

- **First** flip after:
  - 4 000 000 hammers in 270 ms
### FPGA Experiments - Distance 2

<table>
<thead>
<tr>
<th>Role</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>Far Aggressor</td>
<td>$\mathcal{F}_+$</td>
</tr>
<tr>
<td>Near Aggressor</td>
<td>$\mathcal{N}_+$</td>
</tr>
<tr>
<td>Victim</td>
<td>$\mathcal{V}$</td>
</tr>
<tr>
<td>Near Aggressor</td>
<td>$\mathcal{N}_-$</td>
</tr>
<tr>
<td>Far Aggressor</td>
<td>$\mathcal{F}_-$</td>
</tr>
</tbody>
</table>

- **Distance-2**
  - $(\mathcal{F}_+ \rightarrow \mathcal{F}_-)^\infty$
  - Distance two double-sided Rowhammer

- **First** flip after:
  - 4 000 000 hammers in 270 ms
  - *Not* within the refresh windows

---

Andreas Kogler (@0xhilbert)  
Jonas Juffinger (@notimaginary_)
FPGA Experiments - Half-Double

- **Half-Double**
  - \(((\mathcal{F}_+ \rightarrow \mathcal{F}_-)^\beta \rightarrow \mathcal{N}_+ \rightarrow \mathcal{N}_-)^\infty\)
  - **Many** distance-2 accesses with a **few** distance-1 accesses

<table>
<thead>
<tr>
<th>Far Aggressor</th>
<th>(\mathcal{F}_+)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Near Aggressor</td>
<td>(\mathcal{N}_+)</td>
</tr>
<tr>
<td>Victim</td>
<td>(\mathcal{V})</td>
</tr>
<tr>
<td>Near Aggressor</td>
<td>(\mathcal{N}_-)</td>
</tr>
<tr>
<td>Far Aggressor</td>
<td>(\mathcal{F}_-)</td>
</tr>
</tbody>
</table>
FPGA Experiments - Half-Double

- **Half-Double**
  - \( ((\mathcal{F}_+ \rightarrow \mathcal{F}_-)^\beta \rightarrow \mathcal{N}_+ \rightarrow \mathcal{N}_-)^\infty \)
  - **Many** distance-2 accesses with a **few** distance-1 accesses
- **First** flip after:
  - 296,960 hammers in 20 ms
  - **Dilution** \( \beta = 57 \) (5120 distance-1 accesses)

<table>
<thead>
<tr>
<th>Role</th>
<th>Notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Far Aggressor</td>
<td>( \mathcal{F}_+ )</td>
</tr>
<tr>
<td>Near Aggressor</td>
<td>( \mathcal{N}_+ )</td>
</tr>
<tr>
<td>Victim</td>
<td>( \mathcal{V} )</td>
</tr>
<tr>
<td>Near Aggressor</td>
<td>( \mathcal{N}_- )</td>
</tr>
<tr>
<td>Far Aggressor</td>
<td>( \mathcal{F}_- )</td>
</tr>
</tbody>
</table>
FPGA Experiments - Half-Double

- **Half-Double**
  - \(((\mathcal{F}_+ \rightarrow \mathcal{F}_-)\beta \rightarrow \mathcal{N}_+ \rightarrow \mathcal{N}_-)^\infty\)
  - Many distance-2 accesses with a few distance-1 accesses

- **First** flip after:
  - 296,960 hammers in 20 ms
  - Dilution \(\beta = 57\) (5120 distance-1 accesses)

- **Within** the refresh window

<table>
<thead>
<tr>
<th>Far Aggressor</th>
<th>(\mathcal{F}_+)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Near Aggressor</td>
<td>(\mathcal{N}_+)</td>
</tr>
<tr>
<td>Victim</td>
<td>(\mathcal{V})</td>
</tr>
<tr>
<td>Near Aggressor</td>
<td>(\mathcal{N}_-)</td>
</tr>
<tr>
<td>Far Aggressor</td>
<td>(\mathcal{F}_-)</td>
</tr>
</tbody>
</table>
FPGA Experiments - Half-Double

- **Half-Double**
  - $((\mathcal{F}_+ \to \mathcal{F}_-)^\beta \to \mathcal{N}_+ \to \mathcal{N}_-)^\infty$
  - Many distance-2 accesses with a few distance-1 accesses

- **First** flip after:
  - 296,960 hammers in 20 ms
  - Dilution $\beta = 57$ (5120 distance-1 accesses)

- **Within** the refresh window
- **Assisted** by TRR

<table>
<thead>
<tr>
<th>Far Aggressor</th>
<th>$\mathcal{F}_+$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Near Aggressor</td>
<td>$\mathcal{N}_+$</td>
</tr>
<tr>
<td>Victim</td>
<td>$\mathcal{V}$</td>
</tr>
<tr>
<td>Near Aggressor</td>
<td>$\mathcal{N}_-$</td>
</tr>
<tr>
<td>Far Aggressor</td>
<td>$\mathcal{F}_-$</td>
</tr>
</tbody>
</table>

Andreas Kogler (@0xhilbert) Jonas Juffinger (@notimaginary_)
FPGA Experiments - Half-Double

- **Half-Double**
  - $((F_+ \rightarrow F_-)^\beta \rightarrow N_+ \rightarrow N_-)^\infty$
  - Many distance-2 accesses with a few distance-1 accesses

- **First** flip after:
  - 296,960 hammers in 20 ms
  - Dilution $\beta = 57$ (5120 distance-1 accesses)
  - Within the refresh window
  - Assisted by TRR

- Attacker $\rightarrow F$

<table>
<thead>
<tr>
<th>Far Aggressor</th>
<th>$(F_+)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Near Aggressor</td>
<td>$(N_+)$</td>
</tr>
<tr>
<td>Victim</td>
<td>$(\mathcal{V})$</td>
</tr>
<tr>
<td>Near Aggressor</td>
<td>$(N_-)$</td>
</tr>
<tr>
<td>Far Aggressor</td>
<td>$(F_-)$</td>
</tr>
</tbody>
</table>

Andreas Kogler (@0xhilbert)  Jonas Juffinger (@notimaginary_)
FPGA Experiments - Half-Double

- **Half-Double**
  - \(((F_+ \rightarrow F_-)^\beta \rightarrow N_+ \rightarrow N_-)^\infty\)
  - **Many** distance-2 accesses with a **few** distance-1 accesses

- **First** flip after:
  - 296,960 hammers in 20 ms
  - Dilution \(\beta = 57\) (5120 distance-1 accesses)
  - ✓ **Within** the refresh window
  - ✓ **Assisted** by TRR

- **Attacker** \(\rightarrow F\)
- **TRR** \(\rightarrow N\)

<table>
<thead>
<tr>
<th>Far Aggressor</th>
<th>((F_+)^\infty)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Near Aggressor</td>
<td>((N_+)^\infty)</td>
</tr>
<tr>
<td>Victim</td>
<td>((V)^\infty)</td>
</tr>
<tr>
<td>Near Aggressor</td>
<td>((N_-)^\infty)</td>
</tr>
<tr>
<td>Far Aggressor</td>
<td>((F_-)^\infty)</td>
</tr>
</tbody>
</table>

Andreas Kogler (@0xhilbert)  Jonas Juffinger (@notimaginary_)
Exploitable in the Wild?
End-to-End Exploit - Overview

- Target PFN in Page Table Entry [3]
End-to-End Exploit - Overview

- Target PFN in Page Table Entry [3]
- C1: Allocation of Contiguous Memory
End-to-End Exploit - Overview

- Target PFN in Page Table Entry [3]
- **C1**: Allocation of Contiguous Memory
- **C2**: Alternative to Memory Templating
End-to-End Exploit - Overview

- Target PFN in Page Table Entry [3]
- C1: Allocation of Contiguous Memory
- C2: Alternative to Memory Templating
- C3: Memory Massaging

Andreas Kogler (@0xhilbert)  Jonas Juffinger (@notimaginary_)
End-to-End Exploit - Overview

- Target PFN in Page Table Entry [3]
- C1: Allocation of Contiguous Memory
- C2: Alternative to Memory Templating
- C3: Memory Massaging
- C4: Bit-Flip Verification
• Mapping from virtual to physical addresses

\[ X_0 = b_8 \]
\[ X_1 = b_{12} \oplus b_{16} \]
\[ X_2 = b_{13} \oplus b_{17} \]
\[ X_3 = b_{14} \oplus b_{18} \]
C1 - Allocation of Contiguous Memory

- Mapping from virtual to physical addresses
- DRAM addressing function
- Mapping physical address to 16 DRAM banks

\[
\begin{align*}
X_0 &= b_8 \\
X_1 &= b_{12} \oplus b_{16} \\
X_2 &= b_{13} \oplus b_{17} \\
X_3 &= b_{14} \oplus b_{18}
\end{align*}
\]
C1 - Allocation of Contiguous Memory

- Mapping from virtual to physical addresses
- DRAM addressing function
- Mapping physical address to 16 DRAM banks
- **Specific** bank access pattern if contiguous memory

\[ X_0 = b_8 \]
\[ X_1 = b_{12} \oplus b_{16} \]
\[ X_2 = b_{13} \oplus b_{17} \]
\[ X_3 = b_{14} \oplus b_{18} \]
C1 - Allocation of Contiguous Memory

\[ X_0 = b_8 \]
\[ X_1 = b_{12} \oplus b_{16} \]
\[ X_2 = b_{13} \oplus b_{17} \]
\[ X_3 = b_{14} \oplus b_{18} \]

- Mapping from virtual to physical addresses
- DRAM addressing function
- Mapping physical address to 16 DRAM banks
- \textbf{Specific} bank access pattern if contiguous memory

✓ Extract pattern with a timing side channel
C2 & C3 - Memory Templating & Memory Massaging

- Skip templating

<table>
<thead>
<tr>
<th>F+</th>
<th>N+</th>
<th>ν</th>
<th>N-</th>
<th>F-</th>
<th>F+</th>
<th>N+</th>
<th>ν</th>
<th>N-</th>
<th>F-</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
</tr>
</tbody>
</table>

Andreas Kogler (@0xhilbert) Jonas Juffinger (@notimaginary_)
<table>
<thead>
<tr>
<th>$\mathcal{F}_+$</th>
<th>$\mathcal{N}_+$</th>
<th>$\nu$</th>
<th>$\mathcal{N}_-$</th>
<th>$\mathcal{F}_-$</th>
<th>$\mathcal{F}_+$</th>
<th>$\mathcal{N}_+$</th>
<th>$\nu$</th>
<th>$\mathcal{N}_-$</th>
<th>$\mathcal{F}_-$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
</tr>
</tbody>
</table>

- Skip templating
- Spray page tables
C2 & C3 - Memory Templating & Memory Massaging

- Skip templating
- Spray page tables
- Hammer with Half-Double
• Corrupt page table entries can **kill** the attacker process
C4 - Bit-Flip Verification

- Corrupt page table entries can kill the attacker process

```c
if ( /*misprediction*/ ) {
    access(probe + (*ptr & 1));
}
if ( is_cached(probe) ) {
    // ptr[0-4] valid
}
```

- Verify if address save to access
- Spectre gadget
C4 - Bit-Flip Verification

• Corrupt page table entries can kill the attacker process

```c
if ( /*misprediction*/ ) {
    access(probe + (*ptr & 1));
}
if ( is_cached(probe) ) {
    // ptr[0-4] valid
}
```

• **Verify** if address save to access
• **Spectre** gadget
• Cached → accessible
C4 - Bit-Flip Verification

- Corrupt page table entries can **kill** the attacker process

```c
if (/*misprediction*/ ) {
    access(probe + (*ptr & 1));
}
if (is_cached(probe)) {
    // ptr[0-4] valid
}
```

- **Verify** if address save to access
- **Spectre** gadget
- Cached $\rightarrow$ accessible
- **Suppresses** corruption faults

Andreas Kogler (@0xhilbert) Jonas Juffinger (@notimaginary_)
End-to-End Exploit - Timings

- **C1**
  - 10s ... 4m

- **C3**
  - < 1m

- **C2**
  - ≈ 23m

- **C4**
  - ≈ 22m
  - ≈ 11m

- **root**

- **45 minutes (Chromebook)**

Andreas Kogler (@0xhilbert)  Jonas Juffinger (@notimaginary_)
End-to-End Exploit - Timings

- **C1**: 10s ... 4m
- **C3**: < 1m
- **C2**: ≈23m
- **C4**: ≈22m
- **C4**: ≈11m
- **root**

- **45 minutes** (*Chromebook*)
- **Full memory read & write primitive**

Andreas Kogler (@0xhilbert)  Jonas Juffinger (@notimaginary.)
End-to-End Exploit - Timings

- **C1** (10s ... 4m)
- **C3** (< 1m)
- **C2** (≈23m)
- **C4** (≈22m)
- **C4** (≈11m)
- **root**

- **45 minutes** (*Chromebook*)
- Full memory read & write primitive
- Deployable *inside* an APP

Andreas Kogler (@0xhilbert)  Jonas Juffinger (@notimaginary)
Final Remarks

- Open Source GitHub: https://github.com/IAIK/halfdouble

Passed artifact evaluation

More details

Dance-experiments

Contiguous memory Z3 solver

Physical address bit recovery

...
Final Remarks

- **Open Source** 🌐 https://github.com/IAIK/halfdouble
- **Passed** artifact evaluation

Andreas Kogler (@0xhilbert)  Jonas Juffinger (@notimaginary_)
Final Remarks

- **Open Source** 🌐 [https://github.com/IAIK/halfdouble](https://github.com/IAIK/halfdouble)
- **Passed** artifact evaluation

- **More** details
  - Dance-experiments
  - Contiguous memory Z3 solver
  - Physical address bit recovery
  - …
Final Remarks

- **Open Source** 📚 [https://github.com/IAIK/halldouble](https://github.com/IAIK/halldouble)
- **Passed** artifact evaluation
- **More** details
  - *Dance*-experiments
  - Contiguous memory: Z3 solver
  - Physical address bit recovery
  - …


Additonal Slides
Affected Devices

- Tested **13** DIMMs & devices
- **2** DIMMs affected
  - FPGA analysis
  - Exact numbers
- **5 out of 7** mobile devices affected
  - Reversed addressing
  - Unprivileged flush
  - Uncachable memory (10x)

Andreas Kogler (@0xhilbert) Jonas Juffinger (@notimaginary_)
## Affected Devices - Flip Numbers

<table>
<thead>
<tr>
<th>System</th>
<th>RAM</th>
<th>$N_{Hammers}$</th>
<th>$UC_{0→1}$</th>
<th>$UC_{1→0}$</th>
<th>$Flush_{0→1}$</th>
<th>$Flush_{1→0}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chromebook₁</td>
<td>LPDDR4x</td>
<td>23 274</td>
<td>27</td>
<td>40</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>Chromebook₂</td>
<td>LPDDR4x</td>
<td>23 586</td>
<td>235</td>
<td>2379</td>
<td>12</td>
<td>101</td>
</tr>
<tr>
<td>OnePlus 5T</td>
<td>LPDDR4x</td>
<td>25 687</td>
<td>2</td>
<td>30</td>
<td>1</td>
<td>24</td>
</tr>
<tr>
<td>Pixel 3</td>
<td>LPDDR4x</td>
<td>32 921</td>
<td>11</td>
<td>5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>HTC U11</td>
<td>LPDDR4x</td>
<td>21 840</td>
<td>-</td>
<td>-</td>
<td>3</td>
<td>17</td>
</tr>
</tbody>
</table>